



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Soustava senzorů a čidel pro snímání veličin na robotické multikoptéře

Bakalářská práce

Studijní program: B2612 – Elektrotechnika a informatika
Studijní obor: 2612R011 – Elektronické informační a řídicí systémy
Autor práce: **Lubomír Pomezný**
Vedoucí práce: Ing. Leoš Petržílka





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

The System of Sensors to Data Capture on Independent Robotic Multicopter

Bachelor thesis

Study programme: B2612 – Electrotechnology and informatics
Study branch: 2612R011 – Electronic Information and Control Systems
Author: **Lubomír Pomezný**
Supervisor: Ing. Leoš Petržílka



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Lubomír Pomezný**
Osobní číslo: **M11000169**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronické informační a řídicí systémy**
Název tématu: **Soustava senzorů a čidel pro snímání veličin na robotické multikoptěře**
Zadávací katedra: **Ústav informačních technologií a elektroniky**

Z á s a d y p r o v y p r a c o v á n í :

1. Nastudujte problematiku autonomního řízení nezávislých robotů se zaměřením na více-rotorová létající zařízení.
2. Pro existující konstrukci již realizované quadcoptery specifikujte senzory vhodné ke snímání provozních parametrů zařízení a sběru dat potřebných k identifikaci okolí.
3. Vyberte a navrhňte soustavu senzorů vhodnou k realizaci v rozsahu bakalářské práce. Zohledněte reálnou dostupnost komponent a ekonomickou únosnost v rámci projektu.
4. Realizujte funkční vzor svého návrhu.
5. Proveďte provozní testy a měření klíčových parametrů navrženého systému.

Rozsah grafických prací: Dle potřeby dokumentace

Rozsah pracovní zprávy: cca 30 stran

Forma zpracování bakalářské práce: tištěná/elektronická

Seznam odborné literatury:

- [1] VLČEK, Jiří. Měření elektrických veličin. dotisk vyd. Praha: Jiří Vlček, 2002, 25 s.
- [2] BERAN, Vlastimil, Josef GIRG a Olga TŮMOVÁ. Měření neelektrických veličin. vyd. Plzeň: Západočeská univerzita, 1994, 189 s. ISBN 80-708-2158-2.

Vedoucí bakalářské práce:

Ing. Leoš Petržílka

Ústav informačních technologií a elektroniky


Konzultant bakalářské práce:

prof. Ing. Zdeněk Plíva, Ph.D.


Ústav informačních technologií a elektroniky

Datum zadání bakalářské práce: 12. září 2014

Termín odevzdání bakalářské práce: 15. května 2015


prof. Ing. Václav Kopecký, CSc.
děkan




prof. Ing. Zdeněk Plíva, Ph.D.
vedoucí ústavu

V Liberci dne 12. září 2014

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 4. 1. 2016

Podpis:



Abstrakt

Tématem bakalářské práce je specifikovat množinu senzorů vhodných k aplikaci na robotickou multikoptéru. Cílem práce je porovnat využití různých čidel a rozhodnout, zda je jejich umístění na multikoptéře vhodné či nikoliv. Práce se zabývá výhradně čidly pro rozšíření možností multikoptéry, nikoliv čidly určenými pro avioniku. Práce porovnává různé vlastnosti čidel ve snaze zjednodušit jejich výběr a použití pro budoucí autonomní mód vybrané multikoptéry.

V druhé části se práce zabývá možnostmi využití dat z akcelerometru pro určení polohy multikoptéry. Tato metoda využívá dvou po sobě jdoucích integrací výstupního signálu akcelerometru. Součástí práce je praktické řešení a rozbor různých možností realizace této metody.

Poslední část této práce se zabývá praktickou realizací metody dvojité integrace zrychlení za účelem výpočtu uražené dráhy multikoptéry. Její součástí tvoří popis jednotlivých částí realizovaného programu včetně navrhnutého a vyzkoušeného alternativního řešení pro detekci zastavení pohybu. Závěr této kapitoly obsahuje zhodnocení použité metody s naměřenými výsledky. Dosažené výsledky nedosahují, při použití daných technických prostředků požadované úrovně přesnosti: průměrná vypočítaná vzdálenost má od vzdálenosti skutečně uražené odchylku 30 %. Takto velká odchylka je však s největší pravděpodobností zapříčiněna akcelerometrem s nevhodně velkým měřícím rozsahem.

Závěr práce se zabývá zhodnocením a možným využitím čidel pro autonomní režim multikoptéry a nabízí několik návrhů na realizaci autonomního režimu včetně popisu jejich nevýhod a možných komplikací, které mohou nastat při jejich použití.

Klíčová slova: multikoptéra, senzory, autonomní režim, akcelerometr, numerická integrace

Abstract

The theme of this bachelor thesis is to determine the applicability of different types of sensors for use on a robotic multicopter. The aim of the thesis is to compare the use of various sensors and decide whether it is appropriate to use them on multicopter or not. The work deals with sensors for operational physical quantities, there is no measurement for avionics quantities. This thesis compares different aspects of sensors to simplify their selection for use in future autonomous mode of the multicopter.

The next part of thesis deals with the possibility of using data from the accelerometer to determine the position. This method uses two consecutive integration of the accelerometer output signal. Part of the thesis is a practical solution and analysis of different options for this method.

The last part deals with the practical implementation of methods for double integration of acceleration to calculate the covered distance. It includes a description of the various parts of a given program, including the proposed and tested alternative solution for detecting end of the movement. At the end of this chapter there is evaluation of used methods with measured results. Achieved results are very imprecise, and the average distance that is calculated differs from actual traveled distance by 30 %. This large deviation, however, is most likely due to the accelerometer unnecessarily large measuring range.

End of the thesis deals with the evaluation and possible use of sensors for autonomous mode of the multicopter and offers some suggestions on the implementation of the autonomous mode, including a description of their disadvantages and possible complications.

Keywords: multicopter, sensors, autonomous mode, accelerometer, numerical integration

Poděkování

Děkuji vedoucímu práce Ing. Leoši Petržílkovi za vedení, cenné rady a podněty a hlavně za trpělivost během tvorby této práce.

Dále děkuji své rodině za podporu během celé doby mého studia.

Obsah

Úvod	12
1 Rešerše senzorů	14
1.1 Ultrazvukové senzory	14
1.2 Barometrické senzory	15
1.3 GPS moduly	16
1.4 Akcelerometry a gyroskopy	16
1.5 Konkrétní senzory	17
1.5.1 HC - SR04	17
1.5.2 HDPM01	18
1.5.3 LPS331AP	18
1.5.4 MPL115A1	18
1.5.5 GPS01XN934A	19
1.5.6 Arduino TinkerKit Accelerometer	19
1.5.7 LIS3DSH	19
1.5.8 MMA7455L	20
1.5.9 MMA8653FC	20
1.5.10 INEMO-M1	20
2 Využití akcelerometru pro sledování dráhy	21
2.1 Teoretický rozbor	21
2.2 Předpoklady pro realizaci	22
2.3 Postup měření	24
2.4 Metody numerické integrace	24
2.4.1 Obdélníková metoda	25
2.4.2 Lichoběžníková metoda	26
2.4.3 Simpsonova metoda	27
2.5 Detekce pohybu a zastavení	28
3 Praktická část	29
3.1 Čtení dat a jejich zpracování	29
3.2 Vzorkování signálu	30
3.3 Zpracování a úprava dat	31
3.4 Detekce zastavení	31
3.5 Integrace	32

3.6	Vyhodnocení	32
	Závěr	35
	Přílohy	38
	A Obsah přiloženého CD	38
	B Ukázka kódu	39

Seznam tabulek

3.1	Naměřená hodnota při dráze 30 cm	33
3.2	Naměřená hodnota při dráze 100 cm	33

Seznam obrázků

2.1	Vývojová deska Arduino UNO	22
2.2	Arduino TinkerKit Accelerometer	23
2.3	Arduino TinkerKit Sensor Shield	23
2.4	Obdélníková metoda numerické integrace	25
2.5	Lichoběžníková metoda numerické integrace	26
2.6	Simpsonova metoda numerické integrace	27

Úvod

Bakalářská práce se zabývá využitím senzorů a čidel na robotické multikoptéře. Hlavním cílem bakalářské práce je zhodnotit využitelnost různých čidel a senzorů k provozním aplikacím na multikoptéře. Jelikož je multikoptéra funkční a plně schopná provozu, bude se tato bakalářská práce věnovat výhradně sensorům sloužícím pro rozšíření schopností vybrané multikoptéry a ne avionice zařízení jako takové. Práce se problematice senzorů vhodných pro multikoptéru věnuje obecně, s hlavním důrazem na typ senzorů, které lze použít k autonomnímu chodu dané multikoptéry. Při shrnutí vybraných senzorů práce porovnává jejich základní parametry, pomocí kterých určuje vhodnost využití při zadané aplikaci. Problematice senzorů se práce věnuje jak z hlediska konkrétního využití daného druhu senzoru, tak i z pohledu zhodnocení konkrétních senzorů, které byly pro tento účel vytipovány a zakoupeny. Práce se snaží zohlednit i ekonomické hledisko multikoptéry a cílí na nejnižší možnou úroveň nákladů.

Druhá část práce se zabývá možnostmi využití akcelerometrů ke zjištění uražené vzdálenosti ve snaze přiblížit multikoptéru autonomnímu režimu. Součástí této kapitoly je teoretická analýza dané problematiky. Dále tato kapitola obsahuje přípravu pro praktické řešení úlohy. Jsou zde popsány i základní možnosti řešení pro numerickou integraci a detekci zastavení pohybu.

Poslední část práce se zabývá konkrétním praktickým výstupem vycházejícím z navrhovaného řešení a popisuje, jak bylo dosaženo výsledků měření a dále tyto výsledky vyhodnocuje. Na tomto místě jsou popsány části programu, který byl

použit pro výsledná data. V neposlední řadě zde dochází k popisu, proč byly zvoleny právě takové prostředky, jaké byly použity. Tato část práce se zabývá pouze tím, zda lze použít data z akcelerometru pro zjištění rychlosti a polohy a jaké lze při daném způsobu měření dosáhnout přesnosti. Součástí této práce není konkrétní implementace tohoto zařízení na multikoptéře, ale zhodnocení dané metody a případná doporučení pro budoucí řešitele konkrétní implementace.

1 Rešerše senzorů

Tato kapitola bakalářské práce se zabývá možným využitím různých druhů senzorů na robotické multikoptéry. Multikoptéra je již postavena, a proto není třeba zabývat se senzory, které jsou důležité pro samotný let multikoptéry. Tato část práce se věnuje nejprve různým druhům senzorů obecně a jejich možnému využití na multikoptéry. Dále popisuje a porovnává již konkrétní senzory, které byly předem zakoupené a tím pádem i fakticky ověřitelné. Z relativně velkého množství senzorů jsou však vybrány jen ty, které by mohly mít praktické využití a jejichž použití má vzhledem k aplikaci smysl. Opomenuty pro svou nevhodnost byly například senzory na měření krátkých vzdáleností (v jednotkách cm), protože jejich využití například pro detekci překážky nemá vzhledem k jejich malému dosahu v dané aplikaci smysl.

1.1 Ultrazvukové senzory

Ultrazvukové senzory fungují na principu vyslání impulsu vysílačem, jeho následném odrazu od detekované plochy a následném zachycení a zpracování impulsu přijímačem. Doba od vyslání pulzu k jeho přijetí (tzv. echo) odpovídá vzdálenosti předmětu od senzoru. Ultrazvukové senzory lze použít i pro zjištění tvaru předmětu nebo ve funkci optické závory. Tyto aplikace se však nehodí pro využití na multikoptéry, a proto se jim práce dále nebude věnovat. Pro detekci překážek ve všech směrech od multikoptéry se jako vhodné jeví využití ultrazvukových senzorů. Při takovéto aplikaci však musí být zohledněno, že senzory budou vystaveny provozním

vlivům multikoptéry. Jedná se jak o možné vibrace, které mohou ovlivnit přesnost měření, tak o charakteristický náklon multikoptéry při běžném provozu (letu), který může způsobovat chybné výsledky vyhodnocování. Při použití ultrazvukového senzoru pro detekci překážek před multikoptérou je tedy nutné brát v úvahu náklon multikoptéry, který může způsobit chybné vyhodnocení překážky. Jedná se zejména o falešné vyhodnocení překážky při letu multikoptéry vpřed. Náklon multikoptéry způsobí, že senzor nesměruje vodorovně a může tak dojít například k vyhodnocení nízkého předmětu jako překážky ve směru letu multikoptéry, i když o překážku v aktuální výšce nejde. Jelikož náklon závisí na aktuální rychlosti, nelze tento problém řešit pevným vyosením senzoru. S podobným problémem je nutné počítat i při použití ultrazvukového senzoru pro detekci výšky, ve které se multikoptéra nachází. V tomto případě ale dochází k jiné chybě, kdy náklon multikoptéry způsobí chybné vyhodnocení výšky - senzor bude vyhodnocovat větší výšku, než ve které se ve skutečnosti nachází.

1.2 Barometrické senzory

Barometrické senzory se využívají k měření atmosférického tlaku. Atmosférický tlak se mění s nadmořskou výškou, a proto je možné využít tento druh senzoru ve funkci výškoměru. Ačkoliv závislost atmosférického tlaku na výšce není v celém svém průběhu lineární, zhruba do výšky 3000 metrů nad mořem se za lineární považuje. Podle fyzikálních tabulek [3] odpovídá změně výšky o 1000 metrů průměrná změna tlaku o 10,4 kPa, z čehož vyplývá, že změně výšky o 1 metr odpovídá změna tlaku o 104 Pa. Dostatečně přesný digitální barometr by tedy jako výškoměr pro multikoptéru šel použít. Nevýhodou tohoto druhu výškoměru je výstup ve formě absolutní hodnoty výšky nad mořem, a ne aktuální výšky nad povrchem. Pro využitelnost jako výškoměru nad povrchem by mohla vyřešit průběžná kalibrace pomocí jiného druhu senzoru, například ultrazvukového. Senzory měřící tlak mohou měřit absolutní tlak, relativní tlak nebo oba předchozí tlaky. Absolutní tlak je tlak,

který je měřený vůči vakuu. Relativní tlak je měřený vůči atmosférickému tlaku, relativní tlak získáme odečtením absolutního tlaku od atmosférického tlaku.

1.3 GPS moduly

Určení polohy pomocí systému GPS (Global Positioning System) je jednou z dalších metod, které lze použít pro určení polohy multikoptéry i ve funkci výškoměru. GPS funguje na bázi příjmu signálů od satelitních družic obíhajících Zemí. Při autonomním provozu je však problémem přesnost GPS, která se pohybuje v řádu metrů a nelze tedy mluvit o dostatečně přesné navigaci. Pro zvýšení přesnosti GPS existují komerční systémy, které pomocí stacionárních stanic na zemi, jejichž poloha je známá, zpřesňují určení polohy zařízení. Tyto systémy GPS (např.: WAAS, EGNOS a QZSS) jsou však nákladné - stojí zhruba dvojnásobek ceny klasického GPS modulu bez zpřesnění polohy.

Při aplikaci tohoto druhu senzoru na multikoptéru se nadále musí brát v potaz, že pomocí GPS navigace se multikoptéra nevyhne terénním překážkám. Tato vlastnost společně s výše zmíněnou nepřesností má za následek nevhodnost samostatného použití tohoto senzoru pro přesný autonomní režim.

1.4 Akcelerometry a gyroskopy

Akcelerometr je senzor, který poskytuje informaci o zrychlení v určeném směru (ose). Data z akcelerometrů jsou velmi důležitá pro avioniku multikoptéry. Ačkoliv se nezdá, že by zrychlení multikoptéry patřilo k provozním informacím, lze akcelerometry teoreticky využít k zaznamenávání pohybu multikoptéry. Jejich měřicí rozsah se většinou udává v násobcích tíhového zrychlení g [m/s^2].

Gyroskop je zařízení určené pro zjištění změny polohy nebo pro zjištění natočení objektu, na kterém je umístěn. Výstupem digitálního gyroskopu je úhlová rychlost ω objektu ve stupních za sekundu ($^{\circ}/s$). Digitální MEMS (Micro-Electro-Mechanical Systems) gyroskopy pracují pomocí využití měření Coriolisovy síly. Stejně jako akcelerometry měří zrychlení v ose, tak gyroskopy měří úhlovou rychlost v ose.

1.5 Konkrétní senzory

Během práce na BP bylo postupně k dispozici několik různých druhů senzorů, které se dají využít na multikoptěře. Následující pasáž se zabývá těmito čidly a senzory a porovnává jejich vlastnosti s ohledem na vhodnost použití pro účely multikoptéry. Vyjma provozních parametrů senzorů a zhodnocení jejich aplikace je vhodné se zabývat i spotřebou součástek, protože je třeba brát v úvahu kapacitu baterie multikoptéry, která je pouze omezeným zdrojem energie. Pro budoucí využití je nutné zohlednit i napájecí napětí použitých součástek, protože při konstrukci soustavy senzorů může být výhodné brát ohled na napájecí napětí, které se v ideálním případě dá shrnout do jedné napěťové úrovně. Dále se v následující části uvádí komunikační rozhraní součástek (pokud je daná součástka podporuje) pro možné zjednodušení budoucího konkrétního návrhu senzorické soustavy.

1.5.1 HC - SR04

Ultrazvukový senzor na měření vzdálenosti objektu. Rozsah měřené vzdálenosti je 2 cm až 400 cm s přesností 3 mm a úhlovým rozsahem 15° . Senzor při přivedení TTL pulzu o délce 10 μs vyšle 8 ultrazvukových pulzů o frekvenci 40 kHz a poté čeká na echo. Podle délky přijmutého pulzu je softwarově vypočítaná vzdálenost objektu, od kterého došlo k odrazu. Minimální měřicí vzdálenost 2 cm způsobená slepou zónou senzoru při použití pro měření výšky či vzdálenosti překážky není relevantní,

protože je možné tuto vlastnost vyřešit umístěním senzoru na multikoptéře. Napájecí napětí senzoru je 5 V s proudovým odběrem 15 mA.

1.5.2 HDPM01

Piezo-rezistivní senzor tlaku s integrovaným kompasem a teploměrem pro měření výšky multikoptéry. Teploměr se v zařízení využívá pro teplotní kompenzaci - tlak je veličina závislá na teplotě a v tomto případě již není nutné používat samostatný senzor teploty. Zařízení podporuje komunikaci po sběrnici I²C. Při měření absolutního tlaku má senzor měřicí rozsah od 30 kPa do 110 kPa. Senzor měří s přesností 10 Pa, což odpovídá přibližně 10 cm při měření výšky pomocí měření atmosférického tlaku. Napájecí napětí modulu je 2,2 V až 3,6 V, proudový odběr je 40 μ A při 50 měřeních za sekundu.

1.5.3 LPS331AP

Digitální barometr s rozsahem 26 kPa až 126 kPa při měření absolutního tlaku pro měření výšky multikoptéry. Tento senzor taktéž obsahuje teploměr, takže disponuje integrovanou teplotní kompenzací. Senzor umožňuje použití vysoce citlivého měřicího módu, kdy senzor měří s přesností 2 Pa, což odpovídá 1,92 cm při měření nadmořské výšky. Při použití tohoto módu je však zvýšený proudový odběr na 30 μ A. Napájecí napětí senzoru je 1,71 V - 3,6 V. Senzor disponuje rozhraním pro komunikaci po SPI a I²C sběrnici.

1.5.4 MPL115A1

Digitální barometr s rozsahem 50 kPa až 115 kPa sloužící pro měření výšky multikoptéry. Jako předchozí barometry taktéž disponuje integrovaným teploměrem pro teplotní kompenzaci. Nevýhodou tohoto senzoru je jeho přesnost - 1 kPa, což

odpovídá přibližně 10 m při měření výšky pomocí atmosférického tlaku. Napájecí napětí senzoru je 2,375 V až 5,5 V s odběrem 5 μ A při jednom měření za vteřinu. Senzor disponuje rozhraním pro komunikaci po SPI sběrnici.

1.5.5 GPS01XN934A

GPS modul, který je navržen pro použití s anténou se ziskem 20 - 30 dB pro určení polohy a výšky multikoptéry. Senzor poskytuje aktuální polohu s přesností na 5 m, rychlost s přesností 0,1 m/s a aktuální čas s přesností 0,1 μ s. Senzor je schopný poskytovat data s frekvencí 1 Hz. Napájecí napětí je 3,3 V a odběr senzoru je 50 mA.

1.5.6 Arduino TinkerKit Accelerometer

Digitální akcelerometr pro připojení k procesorové desce Arduino měřící zrychlení multikoptéry. Jedná se o dvouosý akcelerometr s rozsahem měření $\pm 3,5$ g. Tento akcelerometr byl použit pro další část práce a bude podrobněji popsán v následujících kapitolách.

1.5.7 LIS3DSH

16b akcelerometr s možností dynamické změny rozsahu (u některých akcelerometrů se změna rozsahu provádí pomocí HW změny zapojení akcelerometru). Senzor umožňuje měřit v rozsahu ± 2 g ± 4 g ± 6 g ± 8 g a ± 16 g. Akcelerometr podporuje komunikaci po sběrnici SPI nebo I²C. Napájecí napětí senzoru je 1,71 V až 3,6 V.

1.5.8 MMA7455L

Akcelerometr s možností změny rozsahu. 10b mód umožňuje měření rozsahů ± 2 g a ± 8 g. 8b mód pro rozsahy ± 2 g ± 4 g a ± 8 g. Senzor podporuje komunikaci po sběrnících SPI nebo I²C. Napájecí napětí senzoru je 2,4 V až 3,6 V.

1.5.9 MMA8653FC

10b akcelerometr s možností změny rozsahu na ± 2 g, ± 4 g a ± 8 g. Komunikace probíhá po sběrnici I²C. Napájecí napětí je 1,62 V až 3,6 V.

1.5.10 INEMO-M1

Modul kombinující tříosý kompas, tříosý akcelerometr a tříosý gyroskop. Rozsah 16b akcelerometru je ± 2 g ± 4 g ± 8 g a ± 16 g s možností dynamické změny rozsahu. Rozsah gyroskopu je ± 250 °/s ± 500 °/s a ± 2000 °/s s možností dynamické změny rozsahu. Modul podporuje komunikaci po sběrnících SPI, I²C, CAN, USART a USB 2.0.

2 Využití akcelerometru pro sledování dráhy

Jednou z možností pro autonomní chod multikoptéry je využití senzorů zrychlení pro zjištění aktuální rychlosti a následně i pro zjištění uražené dráhy. Tato metoda se stává stále populárnější, ale názory na její využití se liší. Následující části práce se budou zabývat především tím, zda je tato metoda použitelná, jaké lze při jejím použití dosáhnout přesnosti a jaká jsou případná úskalí této aplikace akcelerometrů.

2.1 Teoretický rozbor

Výstupem z akcelerometrů je zrychlení (po přepočtení x-bitového výstupu na hodnotu zrychlení). Ze vzorce 2.1 pro výpočet okamžitého zrychlení:

$$\vec{a} = \frac{d\vec{v}}{dt} \quad (2.1)$$

kde \vec{a} je zrychlení, \vec{v} je rychlost a t je čas, a ze vzorce 2.2 pro výpočet okamžité rychlosti:

$$\vec{v} = \frac{ds}{dt} \quad (2.2)$$

kde s je dráha, lze odvodit vztah 2.3 pro výpočet okamžitého zrychlení pomocí dráhy

$$\vec{a} = \frac{d^2s}{dt^2} \quad (2.3)$$

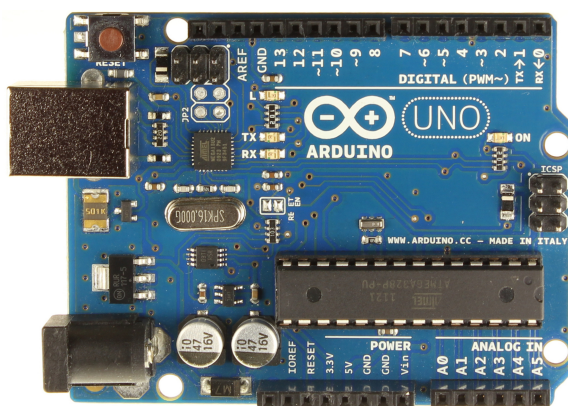
Jelikož je integrace inverzním procesem k derivaci, lze vztah 2.3 upravit do následující podoby:

$$s = \int \left(\int \vec{a} \, dt \right) dt \quad (2.4)$$

pro výpočet dráhy (vzorec 2.4).

2.2 Předpoklady pro realizaci

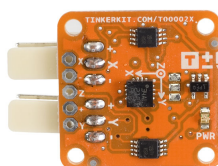
Aby bylo možné tuto metodu vyzkoušet a otestovat, je kromě akcelerometru potřeba i mikropočítač, ke kterému se daný akcelerometr připojí a který bude data z akcelerometru zpracovávat. Z dostupných procesorů byla vybrána vývojová deska ArduinoUno (obr 2.1) osazená procesorem ATmega328. Jedná se o 8bitový AVR procesor s maximální frekvencí 20 MHz (na desce ArduinoUNO je procesor vybaven 16 MHz krystalem). Výhodou zvolené vývojové desky je zabudovaný převodník, který umožňuje posílat program do procesoru pomocí rozhraní USB. Pomocí USB kabelu je celá vývojová deska i s připojenými periferiemi napájena. Program pro procesor se píše v programovacím jazyce Wiring, který vychází z programovacího jazyka C++. Pro psaní programu se používá vývojové prostředí ArduinoIDE, které je zdarma ke stažení na oficiálních stránkách projektu Arduino.



Obrázek 2.1: Vývojová deska Arduino UNO

Zdroj:http://d2rormqr1qwzpz.cloudfront.net/photos/2013/06/12/48912-arduinouno_r3_front.jpg

Jako akcelerometr byl zvolen Arduino TinkerKit T000020 (obr 2.2). Jedná se modul osazený akcelerometrem LIS344AL. Akcelerometr snímá zrychlení ve třech osách, modul má však vyvedeny pouze osy dvě. Třetí osu je možné v případě potřeby vyvést samostatným vodičem. Modul se k vývojové desce připojuje pomocí TinkerKit Sensor Shieldu (obr 2.3), který usnadňuje práci se vstupy a výstupy vývojové desky. Akcelerometr je 10bitový a pracuje v rozsahu ± 3 g, jeho výstup tedy nabývá hodnot 0-1023.



Obrázek 2.2: Arduino TinkerKit Accelerometer

Zdroj:<http://static.rapidonline.com/catalogueimages/Product/S73-4658P02WL.jpg>



Obrázek 2.3: Arduino TinkerKit Sensor Shield

Zdroj:http://www.robotmesh.com/media/catalog/product/cache/1/image/1280x/040ec09b1e35df139433887a97daa66f/t/i/tinkerkit_sensor_shield_01.jpg

Stavebnice Arduino byla vybrána především pro svou jednoduchost, modularitu, dostupnost a možnost rozšíření. Vybráním vývojové desky ArduinoUno v kombinaci s TinkerKitem jako akcelerometrem odpadá nutnost výběru mikrokontroleru, jeho zapojení (krystal, filtrační kondenzátory atd.) a také odpadá nutnost programově

řešit komunikaci akcelerometru s MCU (Microcontroller Unit). Využití Arduina dále ulehčuje i samotný sběr dat z akcelerometru, protože vývojové prostředí pro Arduino má možnost přijímat data z procesoru do Sériového monitoru v reálném čase a odpadá tím nutnost data ukládat, přenášet do PC a poté analyzovat při vývoji programu pro aplikaci.

2.3 Postup měření

Čtení dat je potřeba provádět v pravidelných intervalech kvůli následné integraci zrychlení. Načtená hodnota se musí přepočítat z bitové hodnoty výstupu akcelerometru na zrychlení pomocí následujícího vztahu:

$$zrychlení = \left(\frac{7}{\text{bitový rozsah}} \cdot \text{bitová hodnota} \right) - 3,5 \quad (2.5)$$

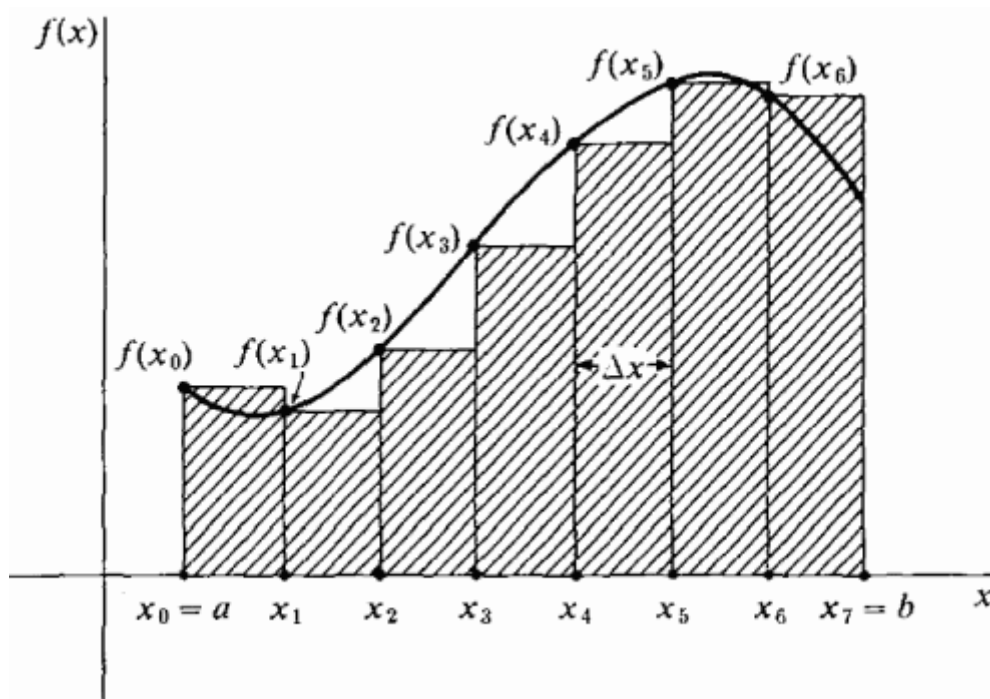
kde 7 je velikost měřicího rozsahu akcelerometru a 3,5 je offset. Hodnotu zrychlení musíme dále analyzovat a zjistit, zda došlo k pohybu nebo ne. Nedošlo-li k pohybu, čekáme na další vzorek. Pokud k pohybu došlo, provedeme integraci vzorku a tím získáme okamžitou rychlost. Tuto hodnotu opět zintegrujeme a získáme tak námi požadovanou polohu. Nakonec uchováme předchozí hodnoty zrychlení, rychlosti a polohy pro využití s dalším vzorkem.

2.4 Metody numerické integrace

Jelikož je pro vypočítání rychlosti a dráhy potřebná integrace, je nutné její výpočet naprogramovat. Výsledkem určitého integrálu je plocha pod křivkou funkce mezi body, které jsou zadány mezemi integrálu. Při řešení určitého integrálu pro digitální signál však nemáme k dispozici předpis funkce, ale jen funkční hodnoty, a proto se provádí aproximace funkce, přičemž dochází ke vzniku určité chyby. Pro aproximování lze použít několik různých metod, jejichž popis následuje.

2.4.1 Obdélníková metoda

Obdélníková metoda předpokládá, že se jedná o funkci, která je po částech lineární. Hodnoty jednotlivých vzorků se tedy mění pouze skokově a s frekvencí odpovídající frekvenci vzorkovací. Předpokládá se, že hodnota vzorku se do dalšího vzorku nemění, a proto tedy stačí sečíst hodnoty všech vzorků vynásobených vzorkovací periodou. Vzorkovací perioda je převrácená hodnota vzorkovací frekvence. Při aplikování této metody v případě použití, kdy data zpracováváme okamžitě, tedy stačí aktuální vzorek vynásobit vzorkovací periodou a přičíst k předchozímu výsledku.



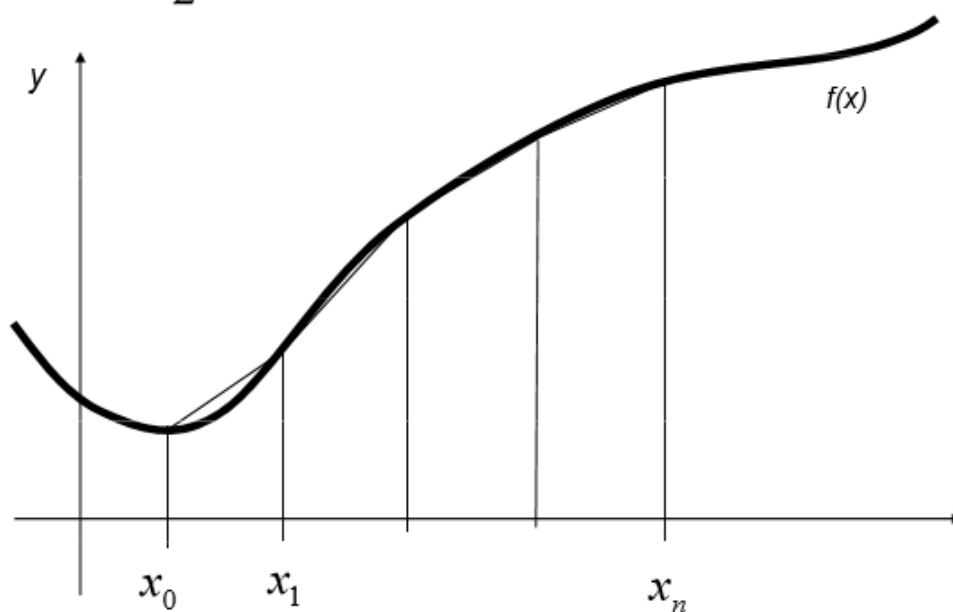
Obrázek 2.4: Obdélníková metoda numerické integrace

Zdroj:http://dailycalculus.blogspot.cz/2008_04_01_archive.html

2.4.2 Lichoběžníková metoda

Tato metoda předpokládá po částech lineární signál. Předpokládá se tedy, že hodnota jednotlivých vzorků se mění lineárně a k výpočtu se využívá plocha pravoúhlého lichoběžníku, kdy naměřené vzorky tvoří jeho rovnoběžné strany. Pokud chceme data zpracovávat v reálném čase, je pro tuto metodu nutné uchovávat předchozí vzorek, který je potřebný pro výpočet.

$$\int_a^b f(x)dx \approx \frac{h}{2}(f(x_0) + f(x_n)) + h(f(x_1) + \dots + f(x_{n-1}))$$

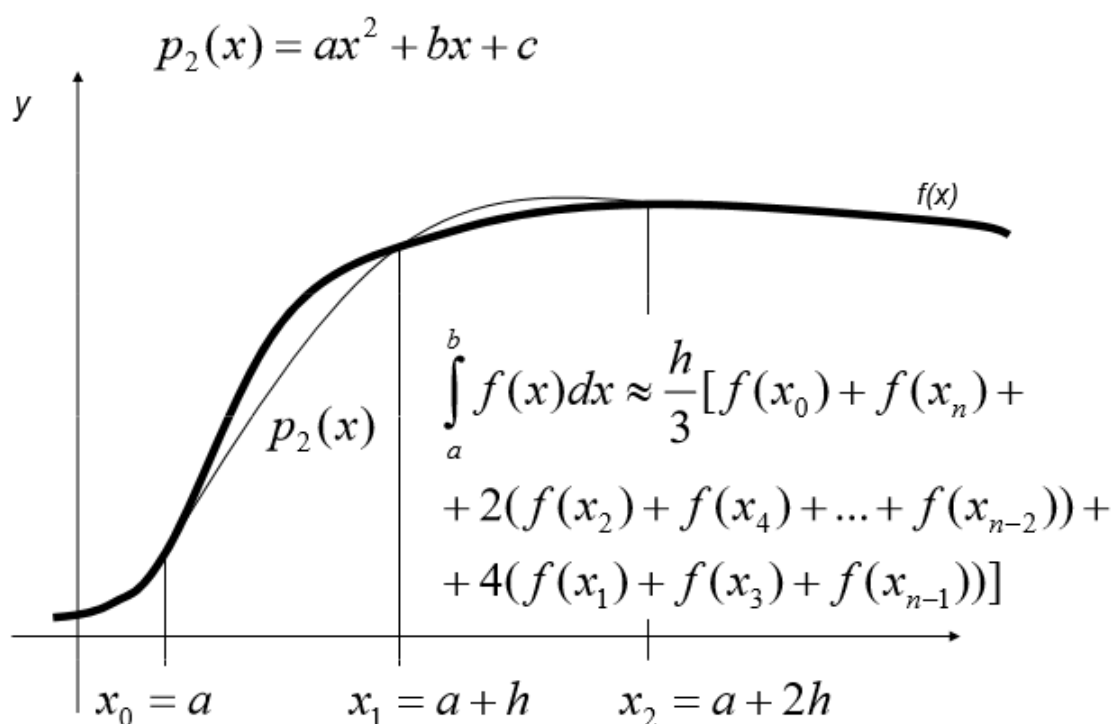


Obrázek 2.5: Lichoběžníková metoda numerické integrace

Zdroj:<http://geometrie.kma.zcu.cz/index.php/www/content/download/950/2681/file/NGM-P4-integrace.pdf>

2.4.3 Simpsonova metoda

Simpsonova metoda předpokládá po částech kvadratickou funkci. Jedná se o zobecnění lichoběžníkové metody o jeden řád interpolace. U této metody je nutné mít k dispozici sudý počet vzorků a znát tři funkční hodnoty funkce, kterou chceme integrovat.



Obrázek 2.6: Simpsonova metoda numerické integrace

Zdroj: <http://geometrie.kma.zcu.cz/index.php/www/content/download/950/2681/file/NGM-P4-integrace.pdf>

2.5 Detekce pohybu a zastavení

Je důležité si uvědomit, že pokud nedochází k pohybu, na výstupu akcelerometru bude nulová hodnota zrychlení respektive polovina jeho 10b rozsahu. Lze předpokládat, že hodnota nebude ustálená přesně na nule, ale bude docházet k jistému šumu. Tento šum lze eliminovat vhodně zvoleným filtrem. Zda došlo k pohybu pak rozeznáme, pokud filtrované zrychlení nabyde jinou než nulovou hodnotu.

Detekce zastavení je však složitější. Pokud bude po započnutí pohybu zrychlení nulové, neznamená to nutně, že k pohybu nedochází. Při nulovém zrychlení může dojít k následujícím dvěma stavům: nedochází k pohybu, nebo dochází k pohybu konstantní rychlostí. Pro detekci zastavení lze použít dvě metody. První metoda spočívá v počítání nulových vzorků zrychlení. Pokud zaznamenáme dostatečný počet nulových vzorků, lze se domnívat, že k pohybu nedochází. Tato metoda je použitelná, pokud lze předpokládat minimální pohyb konstantní rychlostí. Druhá a spolehlivější metoda spočívá ve sčítání zrychlení (pokud se rychlost zvyšuje) a odečítání zrychlení (při snižování rychlosti), dokud se nedosáhne nulové hodnoty. Lze předpokládat, že vlivem vzorkování signálu nemusí dojít k přesně nulové hodnotě, a proto je třeba určit meze, při kterých dojde k vyhodnocení zastavení pohybu.

3 Praktická část

Tato část práce se věnuje konkrétní implementaci výše uvedené metody a jejímu odzkoušení v praxi. Praktická aplikace je provedena nejprve pro jednu osu zrychlení pro odladění chyb a zjištění, zda má smysl přepisovat program pro využití druhé osy. Využití Arduina přináší usnadnění komunikace s akcelerometrem. Další vlastností Arduina je nahrazení funkce *main()* funkcí *loop()*, ve které se odehrává vlastní řešení aplikace. Tato funkce se cyklicky opakuje a přibližuje programování Arduina spíše programování PLC zařízení.

3.1 Čtení dat a jejich zpracování

Prvním úkonem v programu je čtení dat z akcelerometru. Akcelerometr komunikuje s procesorem pomocí sériového rozhraní USART. Pro zajištění zahájení seriové komunikace je nutné v inicializační funkci programu *setup()* nastavit započnutí komunikace a nastavení baud rate pomocí příkazu *Serial.begin(9600);*. Konkrétní data přímo z akcelerometru získáme pomocí funkce *analogRead()*. Návrátovou hodnotou této funkce je bitová hodnota na pinu, který je uveden v argumentu funkce. Data z akcelerometru se ukládají do proměnné *senzorX* pro osu X.

Hodnota z akcelerometru se ukládá do proměnné *sensorX*. Po načtení hodnoty do proměnné, je třeba bitovou hodnotu přepočítat na zrychlení. Hodnota zrychlení se vypočítá pomocí dostupných dat z katalogového listu součástky. Jelikož je znám měřící rozsah součástky, linearita a rozsah výstupu akcelerometru [4], lze zjistit

velikost kvantizační hladiny. Tato hodnota pak stačí vynásobit přijmutou hodnotou z akcelerometru a provést offset o polovinu rozsahu akcelerometru. Výsledný vzorec tedy vypadá následovně:

$$zrychleniX = (((7.0/1024) * sensorX) - 3.5) * g; \quad (3.1)$$

3.2 Vzorkování signálu

Pro potřeby integrování je nutné zajistit relativně přesné časování pro čtení jednotlivých vzorků. Toto časování bylo nejprve realizováno pomocí funkce *delay()*, kde se jako argument funkci předává počet ms. Tato funkce zajišťuje, že se program na zadanou dobu zastaví. Pokud by trvání vykonání jednoho cyklu funkce *loop()* bylo konstantní, šla by tato funkce použít. Doba jednoho cyklu se však může kvůli použitému větvení lišit, a proto je tento postup pro časování nevhodný.

Časování je nakonec realizováno pomocí funkce *millis()*, jejíž návratová hodnota je počet ms od zapnutí programu. Z dokumentace funkce [1] je patrné, že nevýhodou této funkce je přetečení této hodnoty jednou za 50 dní, pro naše potřeby je ale tato funkce dostačující a přesná. V programu se využívá proměnná *casovani*, která slouží jako hodnota, se kterou se porovnává návratová hodnota funkce *millis()*. V případě, že je hodnota funkce vyšší, je splněna podmínka a dojde k vykonání další části programu. Aby podmínka nemusela být realizována pomocí funkce celočíselného zbytku po dělení (modulo), je při splnění podmínky k proměnné *casovani* přičtena hodnota o délce žádané periody vzorkování v ms.

Vzorkovací perioda byla zvolena na hodnotu 30 ms, protože tato vzorkovací frekvence dává dostatečný počet vzorků pro výpočet a procesor má dostatečný čas na vypočítání potřebných hodnot.

3.3 Zpracování a úprava dat

Při sledování přijatých dat v Sériovém monitoru se ukázalo, že v klidovém stavu akcelerometru není na výstupu akcelerometru polovina rozsahu. Proto bylo potřeba upravit rozsah akcelerometru při přepočítání na zrychlení do následujícího tvaru:

$$zrychleniX = (((7.0/(500.0 * 2.0)) * sensorX) - 3.5) * g; \quad (3.2)$$

Kalibrace akcelerometru je podle jeho katalogového listu [4] prováděna při výrobě, a proto není možné ji dodatečně ať už softwarově, nebo hardwarově upravit.

Dalším zpozorovaným jevem je neustálá změna výstupní hodnoty akcelerometru v klidovém režimu. I když je akcelerometr v klidu, na Sériovém monitoru lze číst hodnoty, které neustále kolísají kolem nulové hodnoty zrychlení. Tento šum je nutné odfiltrovat, protože zanáší chybu do měření špatným vyhodnocením, zda dochází k pohybu či ne. Jako řešení byl použit filtr typu pásmová zadrž, který odfiltruje každé zrychlení, které je v rozmezí $-0,3 - 0,3 \text{ m/s}^2$. Pokud je detekováno zrychlení v tomto intervalu, je zrychlení automaticky nastaveno na hodnotu 0.

3.4 Detekce zastavení

Pro detekci zastavení byla nejprve použita výše zmíněná, přesnější metoda. Tato metoda spočívala ve sčítání a odečítání hodnoty zrychlení, kdy zastavení pohybu bylo detekováno, když se tato hodnota blížila nule. Tato metoda ale bohužel nebyla schopna spolehlivě detekovat zastavení pohybu. V drtivé většině případů detekce zastavení neproběhla, i když byl již akcelerometr v klidu. Nelze samozřejmě předpokládat, že se při odečítání a sčítání zrychlení hodnota vrátí přesně zpět na nulovou hodnotu, ale chyba nastávala i přes nastavené odchylky od nuly (tzv. drift).

Detekce zastavení pohybu je nakonec realizována méně přesnou metodou, která je však spolehlivější. Tato metoda spočívá v počítání vzorků v nule a po několika vzorcích v nule se vyhodnotí jako zastavení pohybu. Nulové zrychlení sice může znamenat konstatní rychlost, ale při námi provedené zkoušce je možnost se konstatní rychlosti vyhnout a tím se vyhnout i chybnému vyhodnocení. Při vyhodnocení dojde k vynulování proměnných rychlosti a zrychlení.

3.5 Integrace

Z vypočítaného zrychlení je potřeba integrací zjistit okamžitou rychlost. Integrace se opakovaně provádí od začátku pohybu do doby, než je detekováno zastavení akcelerometru. Samotný výpočet integrace je prováděn pomocí lichoběžníkové metody. K předchozí hodnotě rychlosti se přičte polovina součtu aktuálního a předchozího vzorku vynásobená periodou vzorkování. Stejný postup je aplikován i pro výpočet uražené dráhy. Výsledný vzorec pro výpočet rychlosti vypadá následovně:

$$rychlostX = rychlostX + (((zrychleniX + zrychleniX_predchozi)/2) * (0.03)); \quad (3.3)$$

Po provedení integrace se aktuální hodnoty zrychlení a rychlosti zapíšou do proměnných, které jsou určené k uchování těchto hodnot pro výpočet s následujícím vzorkem zrychlení.

3.6 Vyhodnocení

Díky již zmíněnému Sériovému monitoru lze výsledky sledovat v reálném čase. Není tedy třeba výsledky a jednotlivé hodnoty ukládat a následně přenášet do počítače na analýzu dat. Pro analýzu dat je do Sériového monitoru posílána informace o bitovém výstupu akcelerometru, zrychlení, rychlosti a uražené dráze.

Z těchto dat lze ihned analyzovat jednotlivé veličiny, jejich přesnost nebo lze snáze odhalit chybu ve výpočtu. Díky této funkci bylo provedeno veškeré odladění použitých funkcí i jejich vyhodnocení (například detekce zastavení).

Samotný výsledek sledování uražené dráhy pomocí zrychlení však není uspokojivý. Při měření předem známé vzdálenosti 30 cm v jedné ose dochází k velkým odchylkám. Velikost odchylky je proměnlivá a podle tabulky 3.1 dosahuje hodnot od 23 % do 56 %. Průměrná hodnota odchylky odpovídá 33 %. Po zopakování pokusu pro větší známou dráhu (100 cm) jsou naměřené odchylky ještě větší (viz tabulka 3.2) s průměrnou odchylkou 40 %, což potvrzuje kumulaci chyby.

Tabulka 3.1: Naměřená hodnota při dráze 30 cm

Pokus	Dráha [cm]	Odchylka [%]	Pokus	Dráha [cm]	Odchylka [%]
1	40	33,0	6	37	23,3
2	45	50,0	7	42	40,0
3	36	20,0	8	33	10,0
4	47	56,7	9	38	26,7
5	38	26,7	10	44	46,7

Tabulka 3.2: Naměřená hodnota při dráze 100 cm

Pokus	Dráha [cm]	Odchylka [%]	Pokus	Dráha [cm]	Odchylka [%]
1	143	43,0	6	165	65,0
2	158	58,0	7	133	33,0
3	122	22,0	8	129	29,0
4	135	35,0	9	147	47,0
5	127	27,0	10	145	45,0

Praktické měření ukázalo, že za námi navozených podmínek nelze vzdálenost určit s dostatečnou přesností, a proto nemá význam testovat další osy akcelerometru. Při použití další osy za stejných podmínek se bude odchylka zvyšovat a přesnost bude menší než v jedné ose. Ve snaze odhalit možnou chybu měření byla využita i jiná osa akcelerometru pro stejné měření. Tato metoda však vykazovala stejné výsledky při měření.

Výše uvedené výsledky však neznamenaají, že je daná metoda nepoužitelná nebo špatná. Při použití citlivějšího akcelerometru s menším rozsahem by mělo být možné dosáhnout větší přesnosti. V těchto podmínkách byla využita jen malá část rozsahu akcelerometru, a proto je vhodnější využít akcelerometr s menším rozsahem než je $\pm 3,5$ g, například akcelerometr s rozsahem ± 1 g. Při využití takového akcelerometru by bylo dosaženo větší přesnosti, ale mělo by být možné použít i přesnější metodu pro detekci zastavení pohybu.

Závěr

Práce se zabývala různými senzory a jejich využitím pro účely autonomního řízení multikoptéry. V první kapitole byly obecně popsány různé druhy senzorů a možnost jejich využití. Z popisu jejich fungování vyplývá, že žádný druh senzorů nelze použít pro autonomní řízení samostatně, ale vždy bude vyžadována kombinace alespoň dvou druhů senzorů. Jako nejvhodnější se pro autonomní režim jeví využití modulu GPS v kombinaci s ultrazvukovými senzory pro detekci překážek. Při této kombinaci může multikoptéra doletět na zadané GPS souřadnice v bezpečné výšce a poté se bude orientovat pomocí ultrazvukových senzorů.

U ultrazvukových senzorů je však nutné vyřešit výše zmíněné problémy, které jsou způsobené samotným principem pohybu multikoptéry. Jednou z možností řešení je zjištění úhlu náklonu pomocí gyroskopu a poté dopočítání skutečné vzdálenosti od povrchu (za předpokladu, že se multikoptéra nachází nad rovnou plochou). Tato metoda je podobná metodě použité v této práci s rozdílem jedné integrace. Výstupem gyroskopu je úhlová rychlost a ke zjištění náklonu je tedy nutné tuto veličinu integrovat. Tato metoda však řeší pouze měření výšky, protože pro detekci překážek kolem multikoptéry je nepoužitelná - hrozí falešné vyhodnocení překážky.

Další možnou metodou řešení je prosté krokování pohybu multikoptéry. V neznámém prostředí by proto multikoptéra v pravidelných časových intervalech zastavila a senzory prozkoumala své okolí. Maximální dosah ultrazvukových senzorů se pohybuje v řádech jednotek metrů. Pro řešení touto metodou je však nutné znát, jakých rychlostí multikoptéra dosahuje, a podle toho určit časový interval měření.

Pro neznámé prostředí však stejně nelze předpokládat, že se bude multikoptéra pohybovat maximální možnou rychlostí.

Praktická část práce dokázala, že pomocí zvoleného hardwaru není možné dosáhnout dostatečně přesných výsledků pro měření uražené vzdálenosti. Při měření dochází k velkým odchylkám, které naprosto znemožňují praktické využití dané metody. Tyto výsledky jsou však pravděpodobně způsobeny nevhodně zvoleným akcelerometrem. Použitý TinkerKit s akcelerometrem LIS344AL má rozsah $\pm 3,5$ g, což je pro naše využití příliš velký rozsah. Při použití citlivějšího akcelerometru by pravděpodobně došlo k přesnějším výsledkům, protože použitý akcelerometr používal jen zlomek svého rozsahu. Vzhledem k možnému využití této metody na multikoptéře byl však zvolen rozsah $\pm 3,5$ g, který se zdál před samotnou realizací měření dostatečný a vybraný s ohledem na možnosti multikoptéry.

Reference

- [1] ARDUINO. *Millis*. [online]. [cit. 2015-08-05]. Dostupné z: <https://www.arduino.cc/en/Reference/Millis>
- [2] BERAN, Vlastimil, Josef GIRG a Olga TŮMOVÁ. *Měření neelektrických veličin*. Vyd. 1. Plzeň: Západočeská univerzita, 1994, 189 s. ISBN 80-7082-158-2.
- [3] BROŽ, Jaromír, ROSKOVEC, Vladimír a VALOUCH, Miloslav. *Fyzikální a matematické tabulky*. 1. vyd. Praha: SNTL - Nakladatelství technické literatury, 1980. 308 s. Řada teoretické lit.
- [4] STMicroelectronics. *LIS344AL Datasheet* [pdf]. [cit. 2015-08-12]. Dostupné z: <http://www.st.ewi.tudelft.nl/~gemund/Courses/In4073/Resources/LIS344AL.pdf>
- [5] VLČEK, Jiří. *Měření elektrických veličin*. 4. dotisk 1. vyd. Praha: BEN, 2002. 25 s.

A Obsah přiloženého CD

- Text BP
 - bakalarska_prace_2016_Lubomir_Pomezny.pdf
 - kopie_zadani_bakalarske_prace_2016_Lubomir_Pomezny.pdf
- Program
 - program_Lubomir_Pomezny.ino

B Ukázka kódu

```
#define IO A0
#define I1 A1
#define g 9.823
const int analogInPin1 = IO;
const int analogInPin2 = I1;
int sensorX = 0;
int sensorY = 0;
int pocitadloX = 0;
bool pohyb = 0;
float zrychleniX = 0;
float zrychleniX_predchozi = 0;
float rychlostX = 0;
float rychlostX_predchozi = 0;
float drahaX = 0;
unsigned long casovani;
int pocitadlo = 0;

void setup() {
  Serial.begin(9600);
  casovani = millis();  inicializaci
}

void loop() {
  if (millis() > casovani) {
    casovani +=30;
```



```

sensorX = analogRead(A0);
//sensorY = analogRead(A1);
zrychleniX = (((7.0/(502.0*2.0))*sensorX) - 3.5)*g;
if (zrychleniX > -0.3 && zrychleniX < 0.3){
zrychleniX = 0;
zrychleniX_predchozi = 0;
}

if (zrychleniX == 0) {
pocitadloX++;
}
else pocitadloX = 0;

if (pocitadloX >= 10){
rychlostX = 0;
rychlostX_predchozi = 0;
pocitadloX = 0;
}

rychlostX = rychlostX + (((zrychleniX+zrychleniX_predchozi)/2)*(0.03));
drahaX = drahaX + (((rychlostX + rychlostX_predchozi)/2)*(0.03));

Serial.print("sensor value: ");
Serial.print(sensorX);
Serial.print(" zrychleni: ");
Serial.print(zrychleniX);
Serial.print(" rychlost: ");
Serial.print(rychlostX);
Serial.print(" draha(cm): ");
Serial.println(drahaX/100);

```

```
rychlostX_predchozi = rychlostX;  
zrychleniX_predchozi = zrychleniX;  
}  
}
```